**TC**

**KOCAELİ ÜNİVERSİTESİ**

**MÜHENDİSLİK FAKÜLTESİ**

**BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**

**Ders Adı : ProgramlamaLabaratuvarı II**

**Proje : Sınav Programı Hazırlama**

**Proje Ortamı: Microsoft Visual Stdio**

**Proje Danışmanı:**

**Arş. Gör. Burcu KIR**

**Hazırlayanlar:**

**110201035 Burak UTKUCU**

**110201053 Feyyaz ÖZHAN**

**110201001 Ramazan GÜÇLÜ**

-İki boyutlu dersler dizisi herbir öğrencinin

bulunduğu sınıfa ait zorunlu derslerini, varsa alttan ve üstten seçmiş

olduğu derslerini tutmaktadır.

İki boyutlu dizinin birinci kolu öğrencileri tutarken ikinci kolu

derslerini tutmaktadır.

int olarak tanımladığımız dersler dizisine öğrencinin seçmiş olduğu

dersler sayısal olarak atanmıştır.

Örneğin ilk öğrencinin, ilk olarak aldığı ders İnkılap Tarihi ise

dersler dizisinin [0][0] elemanı 0 dır.

ilk öğrencinin, ikinci olarak aldığı ders Fizik ise dersler

dizisinin [0][1] elemanı 1 dir.

voidsifirla(int dersler[][25], intkullanici): Bu fonksiyonun iki

parametresi bulunmaktadır.

Kullanıcı olarak tanımladığımız değişken o an itibariyle ders

seçimi yapmakta olan kullanıcının sayısını tutmaktadır. Örneğin ilk

kullanıcı için 1, ikinci kullanıcı için 2 değerini tutmaktadır. Her

kullanıcı derslerini seçtiğinde kullanıcı değişkeninin sayısının

artması gerektiği için static olarak tanımladık.

Sifirla fonksiyonu, kullanıcı dersleri seçmeye başlamadan önce, bütün

derslerinin değerini -1 yapmaktadır. öğrencinin 0. Dersi seçmiş olma

ihtimali olduğundan dolayı derslerin değerini ilk olarak 0 değilde -1

yaptık.

intders\_ata(int dersler[][25]): Öğrenci tarafından seçilecek olan

derslericheckbox kullanarak form ekranına yazmıştık. Bu fonksiyon

öğrencilerin seçmiş olduğu dersleri, checkboxların işaretlenme

durumlarını kontrol ederek dersler iki boyutlu dizisine aktarır.

if(checkBox15->CheckState == CheckState::Checked){dersler[x][y]=10;}

kodu şu işi yapar: öğrenci eğer checkBox15 de bulunan Bilgisayar

Organizasyonu ve Mimarisi dersini seçtiyse,fonksiyon bu dersin

numarası olan 10 sayısını dersler dizisine aktarır.

void kontrol(): Öğrencilerin seçmiş olduğu dersleri checkboxlardan

bularak tek tek sayma işlemini yaptırdık.

Öğrenci 1. sınıfta ise 2. Sınıftan seçmiş olduğu dersleri; 2. Sınıfta

ise 3. Sınıftan; 3.sınıfta ise 4. Sınıftan seçmiş olduğu derslerin

sayısınısayaç görevinde bir değişken tanımlayarak kontrol ettirdik.

Değişkenin değerinin 3 ten büyük olup olmadığına karar verdik. Eğer

sayaç olarak kullandığımız değişken 3 ten fazla ise Hata mesajını

ekrana yazdırdık.Bu arada seçilen dersleri sayma işlemini yaparken,

öğrencinin alttan aldığı derslerin sayısını hesaba katmadık, istediği

kadar alttan ders almasına izin verdik.

Eğer kullanıcının bir üst sınıftan seçmiş olduğu derslerin sayısı 3 ten

fazla değilse sifirla() ve ders\_ata() fonksiyonlarıyla

öğrencinin seçmiş olduğu dersleri checkboxları kullanarak dersler iki

boyutlu dizisine diziye aktardık.

Ders seçimi yapan öğrenci sayısı sayısı 20 ye ulaşmış ise WelchPowel

algoritmasını yapmak üzere ana\_program() adlı fonksiyonumuzu çağırdık.

Ardından ekranda son olarak kalan, ders seçim olarak kullandığımız

checkboxları, checkboxlarınVisible özelliğini kullanarak kaldırdık.

//ekrana yazdırma

voidbasa\_don(int dersler[][25]): Ders seçimi yapan öğrenci sayısı 20

ye ulaşmamış ise birtakım işler yaptırdık.

- checkboxlarınVisible özelliğini kullanarak seçimlik

dersleri form ekranından kaldırdık ve Checked özelliğini kullanarak

secilebilecek dersleri işaretsiz yaptırdık.

-sınıf seçim checkboxlarınıVisible özelliğini kullanarak kaldırdık.

private:System::Void checkBox1\_CheckedChanged(System::Object^ sender,System::EventArgs^e):

öğrenci ders seçimi yaparken 1. Sınıf olduğunu belirten checkbox ı

işaretlerse yapılacak komutları buraya yazdık. Öğrencinin 2., 3., 4.

sınıfları işaretlemesini engellemek için, öğrenci checkBox1 e

tıkladığındacheckboxlarınenable özelliğini kullandık. Aynı kodları

sınıf sayılarını değiştirerek üst sınıf öğrencileri için de yaptık.

private: System::Void button5\_Click(System::Object^ sender, System::EventArgs^ e):

öğrenci tamam butonuna basarsa hata mesajını ve tamam butonunu ekrandan

kaldırma işlemini burada yaptık.

private: System::Void button2\_Click(System::Object^ sender, System::EventArgs^ e):

öğrenci ders ekle butonuna basarsa yapılacak işlemleri buraya yazdık.

Ekranda o anda bulunan labelleri ve checkboxlarıVisible=false

koduyla kaldırdık ve yerine istediğimiz checkboxları ve labelleri

görünmesi için Visible=true koduyla yerlerine yerleştirdik.

Aynı zamanda öğrencinin bir üst sınıf hariç diğer üst sınıflardan ders

seçmesini engellemek için enable=false kodunu kullandık.

Kodların geri kalanının çoğunu form ekranında kullanmış olduğumuz

labellerin, textboxların, checkboxların, buttonların ve son olarak

gerçekleştirdiğimiz tablonun özellikleri vs. oluşturmaktadır.

**anaprogram() fonksiyonu:**

welchpowel renklendirme algoritması bu fonksiyonun içinde yapılmaktadır. Derslerin komşu oldukları dersleri tutmak için ders\_komsuluk[25][25] isimli dizisini tanımladık. Derslerin kaç tane komşu sayısı olduğunu tutmak için komsuluk\_sayisi[25][2] dizisini tanımladık.

ilk\_deger(ders\_komsuluk) fonksiyonuyla ders\_komsuluk dizisine ilk değeri atadık.

For döngüsü ve if koşulu yardımıyla öğrencinin seçtiği dersleri tutan dersler[20][25] dizisini kullanarak her dersin komşu olduğu dersi bulup ders\_komsuluk dizisine atadık.

For döngüsü ve if koşulu yardımıyla ve ders\_komsuluk dizisini kullanarak her dersin kaç komşusu olduğunu bulup komşuluk\_sayisi dizisine atadık. Komşuluk\_sayisi dizisinin 0. Sütunu ders numarasını 1. Sütunu ise o dersin kaç komşusu olduğunu tutmaktadır.

komsuluk\_sayisi\_sirala(komsuluk\_sayisi) fonksiyonuyla komşuluk sayılarını welchpowel algoritmasına göre büyükten küçüğe sıraladık.

For döngüsü ve if koşulu yardımıyla bu komsuluk sayılarına göre renklendirme yaptık. Renkleri renk\_sayisi[25][2] dizisinin 1.sütununda tuttuk. renk\_sayisi dizisinin 0. Sütununda dersin numarasını tuttuk.

**İlk\_deger() fonksiyonu:**

Ders\_komsuluk dizisinin tüm elemanlarına -1 ataması yapar. Bu dizide elemanı -1 olan dersler boş anlamına gelir.

**Varmi() fonksiyonu:**

Öğrencinin seçtiği derslerden komşu dersleri bulurken komşu olan ders ders\_komsuluk dizisine atanmış mı atanmamış mı diye kontrol eder.

**Komsuluk\_sayisi\_sirala() fonksiyonu:**

Öğrencinin seçtiği derslerin komşuluk sayiları bulunup komsuluk\_sayisi dizisine atandıktan sonra bu diziyi büyükten küçüğe sıralar.

**Komsu\_degilse() fonksiyonu:**

Renklendirme yaparken ders\_komsuluk dizisi kullanarak dersin komşusu değilse 1 döndürür.

**Komsusu\_farkli\_renk\_mi() fonksiyonu:**

Renklendirme yaparken dersin komsularını tek tek kontrol ederek komşusunda aynı renk varsa atama yapılamayacağı için 0 döndürür. Eğer aynı renk yoksa atamanın yapılabileceği için 1 döndürür.

**Ekrana yazma işlemi:**

For döngüsü ve if koşulu yardımıyla ve renk\_sayisi dizisini kullanarak renk sayisi aynı olan dersleri ders numarasına göre ders\_listesi dizisinden bulup ilgili labelin içine alt alta ekledik.